Rule DAS621: The number of index buffers should be increased (direct access)

Finding: CPExpert noticed that direct processing accounted for a significant amount of the I/O activity to the index component of VSAM data sets. However, insufficient buffers were assigned to the index component. The number of index buffers should be increased for optimal performance of the VSAM data sets listed. This finding applies only if SMF Type 42 (Data Set Statistics)[1] and SMF Type 64 (VSAM Statistics) records are available in a MXG performance data base.

Impact: This can have a LOW IMPACT, MEDIUM IMPACT, or HIGH IMPACT on the performance of applications referencing the VSAM data. The level of impact depends on the number of direct I/O operations.

Discussion: A VSAM file structure consists of one or more *Control Intervals (CIs)* and one or more *Control Areas (CAs)*.

- A **Control Interval** is a continuous area of direct access storage that VSAM uses to store logical records. When a logical record is read from direct access storage, the entire Control Interval containing the record is read into a VSAM buffer in virtual storage. The desired logical record is then transferred from the VSAM buffer to a user-defined buffer or work area. While logical records within a Control Interval may vary in length, all Control Intervals in a specific VSAM data set are of the same length.

- A **Control Area** contains one or more Control Intervals. The Control Intervals are grouped together into fixed-length contiguous areas of direct access storage. A VSAM data set is composed of one or more Control Areas.

Each VSAM data set is defined as a cluster of one or more components.

- The *data component* is the part of a VSAM data set, alternate index, or catalog that contains the data records. The minimum size of a Control Area for a data component is one track, and the maximum size is one cylinder of DASD storage.

- The *index component* is a collection of logically sequenced keys. A key is a value taken from a fixed defined field in each logical record in the VSAM data set. The key identifies the record's position in the data set. Using the index, VSAM is able to randomly retrieve a record from the

[1] **%LET TYPE42DS = Y;** must be specified in USOURCE(GENGUIDE) must be specified in USOURCE(GENGUIDE) or USOURCE(DASGUIDE) to advise CPExpert that TYPE42DS is available.

data component when a request is made for a record with a certain key. The size of the Control Area for an index component is one track of DASD storage.

Key-sequenced data sets (KSDS) and variable-length relative records data sets (VRRDS) contain both a data component and an index component. Additionally, each alternate index contains a data component and an index component. Entry-sequenced data sets (ESDS), linear data sets (LDS), and fixed-length relative record data sets (RRDS) contain only a data component.

The index component consists of two parts: *sequence set* and *index set*.

- The sequence set is the lowest level of index control intervals and directly points to the data Control Interval in the data Control Area. There is one Control Interval in the sequence set for each *data* Control Area. This *index* Control Interval contains pointers and high key information for each *data* Control Interval. The index Control Interval also contains horizontal pointers from one sequence set Control Interval to the next higher keyed sequence set Control Interval.

- The records in all levels of the index above the sequence set are called the index set. If there is more than one sequence set Control Interval, VSAM automatically builds another index level. An entry in an index set record consists of the highest possible key in an index record in the next lower level, and a pointer to the beginning of that index record.

I/O buffers are used by VSAM to read and write control intervals from DASD to virtual storage. A minimum of one buffer is required for an index Control Interval. Having only one index I/O buffer does not hinder performance when the VSAM data set is accessed sequentially, because VSAM gets to the next Control Interval by using the horizontal pointers in sequence set records rather than the vertical pointers in the index set.

The minimum of one index buffer[2] is inadequate with **direct** access to the VSAM data set. When using direct access to retrieve a record from a key-sequenced data set or variable-length RRDS (or store a record using keyed access), VSAM needs to examine the index of the data set. When an index record must be retrieved to locate a data record, VSAM makes room for the new index record by deleting the index record that VSAM judges to be least useful under the prevailing circumstances. If only one index buffer were provided, a serious performance problem would occur if an index record were continually deleted from virtual storage to make

[2]Multiple buffers for the *data component* do not increase performance with direct processing, because only one data buffer is used for each access.

room for another index record, and then retrieved again later when it is required.

Providing more than the minimum number of index can significantly improve performance. IBM benchmarks[3] show over 50% reduction in EXCPs and almost 50% reduction in CPU time, by increasing the number of index buffers from 1 to 4 for non-shared resource (NSR) with direct access.

Unused index buffers do not normally degrade performance, so an adequate number should be specified. For optimum performance, the number of index buffers should be at least as large as the number of high-level index set Control Intervals, plus one per string to contain the entire high-level index set and one sequence set control interval per string in virtual storage.

After applying the screening criteria specified for VSAM data sets, and extracting SMF Type 64 information for those VSAM data sets, CPExpert examines SMF Type 42 (Data Set Statistics) information for the selected VSAM data sets. CPExpert uses the TYPE42DS information (MXG variable DSTYPE) to identify KSDS and VRRDS VSAM data sets. CPExpert selects these VSAM data sets that were used with NSR (using MXG variable S42DSBUF). CPExpert uses the TYPE42DS information to compute the percent of direct accesses to the VSAM data set, using the following algorithm:

$$Percent\ direct\ accesses = \frac{S42AMDRB}{S42AMSRB + S42AMDRB}$$

where: S42AMSRB = Blocks read using sequential access
S42AMDRB = Blocks read using direct access

CPExpert produces Rule DAS621 under the following conditions:

- The TYPE42DS S42DSBUF variable showed that NSR was used for KSDS or VRRDS VSAM data sets, **and**
- The percent of direct accesses for the index component was greater than **DIRINDEX** guidance variable in USOURCE(DASGUIDE), **and**
- The number of buffers (the MXG BUFDRNO variable) assigned to the index component was less than the number of index levels (the MXG ACCLEVEL variable) for the VSAM data set.

[3] *VSAM Demystified*, SG24-6105, Table 6 (NSR buffering with direct access - STRNO=1)

The default value for the DIRINDEX guidance variable is 25%, so CPExpert will produce Rule DAS621 for NSR VSAM data sets when more than 25% of the accesses were direct for the index component and the number of buffers assigned to the index component was less than the number of index levels.

The following example illustrates the output from Rule DAS621:

```
RULE DAS621:  THE NUMBER OF INDEX BUFFERS SHOULD BE INCREASED

   VOLSER: PRD005.  Non-Shared resources (NSR) was specified as the
   buffering technique for the below VSAM data sets, and most of the
   accesses were random processing.  However, relatively few index
   buffers were assigned to the data sets.  You should consider
   increasing the number of index buffers to the number of index
   levels.  The I/O RATE is for the time the data set was open.
                                                          ..       I/O    -ACCESS TYPE (PCT)-  INDEX    BUFFERS
   SMF TIME STAMP    JOB NAME  VSAM DATA SET              ..       RATE   SEQUENTIAL   DIRECT  LEVELS   ASSIGNED
   10:04,19SEP2002   PGRED01D  PWFWFA.PK.GENTRAN.EDIOEA.INDEX..... 30.7       0.0      100.0      3         1
   10:20,19SEP2002   PFFED01D  PWFWFA.PK.GENTRAN.EDIOEA.INDEX..... 15.7       0.0      100.0      3         1
```

While not shown in the above example, CPExpert also shows the OPEN time for the VSAM data set. This normally is the duration of the current OPEN. If %LET VSAMSMRY=Y; was specified in USOURCE(DASGUIDE), the OPEN time represents the sum of the times the VSAM data was OPEN for all TYPE64 records in the performance data base.

Suggestion: If CPExpert produces Rule DAS621, you should consider the following alternatives:

- **Increase the number of index buffers**. You should consider increasing the number of index buffers allocated for the VSAM data sets listed by Rule DAS621.

- **Consider the use of System Managed Buffering**. If the data sets listed by Rule DAS621 are SMS managed, have extended format, and your installation has DFSMS V1R4 or later, you can use system managed buffering (SMB)[4]. System managed buffering enables VSAM to determine the optimum number of index and data buffers, as well as the type of buffer management (LSR or NSR). IBM benchmarks[5] show up to

[4] Please review *DFSMS: Using Data Sets* (Section 2.5.4.2.3: Processing Guidelines and Restrictions) before implementing system managed buffering.

[5] *VSAM Demystified*, SG24-6105, Table 7 (Direct access: Benefits of using SMF: Updates and insertions)

90% reduction in EXCPs, DASD connect time, and CPU time when converting to system managed buffering!

To indicate that VSAM is to use SMB, specify either of the following options:

- Specify the ACCBIAS subparameter of the JCL DD statement AMP parameter.

- Specify Record Access Bias in the data class and an application processing option in the ACB. The ACB must be NSR and MACRF cannot contain any of the following:

    - ICI (Improved control interval processing)
    - AIX Processing the data set through the alternate index of the path specified in the DDname
    - UBF (Management of I/O buffers left up to the VSAM user)

Regardless of whether the JCL DD statement AMP is used or the data class/ACB is used, you should specify the following for the record access bias:

- **Specify Direct Optimized (DO)** for the record access bias for the record access bias if *direct access is near 100% for the VSAM data sets listed*. The DO processing technique optimizes for totally random record access. This technique overrides the user specification for non-shared resources (NSR) buffering with a local shared resources (LSR) implementation of buffering.

- **Specify Direct Weighted (DW)** for the record access bias if the data sets are processed using a mixture of direct and sequential access, and *direct access is dominate*. DW processing provides the minimum read-ahead buffers for sequential retrieval and the maximum index buffers for direct requests.

- **Specify Sequential Weighted (SW)** for the record access bias if the data sets are processed using a mixture of direct and sequential access, and *sequential access is dominate*. This technique uses read-ahead buffers for sequential requests and provides additional index buffers for direct requests. Approximately 100K of processor virtual storage for buffers is required for this technique, defaulted to above 16 MB.

- **Consider the use of Batch LSR**. For batch applications **and if Rule DAS621 shows that direct processing[6] is near 100%**, you can use Batch LSR. Batch LSR provides advantages in an application using VSAM NSR buffering techniques to switch to LSR without changing the application source code or link-editing the application again. Only a JCL change is required.

- If the above actions are not appropriate, you can change the **DIRINDEX** guidance variable in USOURCE(DASGUIDE). Section 3 describes how to change the DIRINDEX guidance variable if you feel that Rule DAS621 is produced prematurely.

- Alternatively, you can exclude the reported VSAM data sets from analysis. Section 3 describes how to exclude VSAM data sets from analysis. However, you should be aware that no analysis of potential VSAM problems will be performed on data sets that are excluded from analysis.

Reference: *DFSMS: Using Data Sets* (SC26-7339 for OS/390; SC26-7410 for z/OS)
Section 2.5.5.5.1 Data Buffers for Sequential Access Using Nonshared Resources
Section 2.5.4.2: Tuning for System-Managed Buffering

IBM Redbook: *VSAM Demystified* (SG24-6105)
Section 2.6.9 (Buffering options)

[6] Using the Batch LSR subsystem with sequential access could degrade performance rather than improve it. This is because Batch LSR does not provide "read-ahead" capability. The "read-ahead" capability is essential for good performance with sequential access.

